91602
new 82-10

LEGAL NOTICE

This report was prepared by Ontario Hydro and, while effort has been made to assure the accuracy and applicability of the material contained herein, neither Ontario Hydro nor any person acting on its behalf:

(a) makes any warranty, express or implied with respect to the use of any information, apparatus, method or process disclosed in this report or that such use may not infringe privately owned rights; or

(b) assumes any liabilities with respect to the use of, or for damages resulting from the use of, any information, apparatus, method or process disclosed in this report.

COMPLIMENTARY COPY
Please Credit Ontario Hydro Research if Reproduced in Whole or in Part

ontario hydro
research division

VERIFICATION OF VAULT RESPONSE FUNCTION SOLUTION

Arthur Chan and Aneel Advani
Statistical Methods & Analysis Unit
Operations Research Department

Abstract

An independent verification of the vault response function solution for the disposal of nuclear waste was completed. The impulse response functions of the convection-diffusion equations which describe the transport of a single nuclide chain, were calculated using Laplace transform methods. Comparing these solutions with the AECL solutions, they were found to be correct within limits of numerical accuracy with a small number of exceptions.

Work done for
Whiteshell Nuclear Research Establishment
Atomic Energy of Canada Limited
under the COG Nulcear Waste Management Program

job	file	date	report no.
72-702	809.7	January 21, 1991	91-26-K

91-07165/

TABLE OF CONTENTS

		Page
1. INTRODUCTION		1
2. PROBLEM DESCRIPTION AND VERIFICATION METHODOLOGY		2
2.1	The Boundary Value Problems	2
2.2	The Solution Method	2
2.3	The Numerical Inversion	3
2.4	Standard for Comparison	4
3. RESULTS		5
4. CONCLUSION		5
REFERENCES		6
FIGURES		
Figure 1	- Number of Discrepancies vs Number of Cases - Vault Response Function	7
Figure 2	- Number of Discrepancies vs Number of Cases - Source Release Function	8
Figure 3	- AECL Solution vs Talbot Solution for the Vault Response Function - case 15	9
Figure 4	- AECL Solution vs Talbot Solution for the Vault Response Function - case 38	10
Figure 5	- AECL Solution vs Talbot Solution for the Source Release Function - case 261	11
Figure 6	- AECL Solution vs Talbot Solution for the Source Release Function - case 309	12
APPENDICES		
Appendix I	Mathematical Formulae and Test Parameter Values for Response and Source Release Functions	13
Appendix II	Computer Code	18

APPENDICES (Cont'd) **Page**

Appendix III	Results and Tables	29
Table 1	Samples of Difference between AECL and Talbot Results for the Source Release Function	30
Table 2	Samples of Difference between AECL and Talbot Results for the Vault Response Function	31
DISTRIBUTION		last

91601
new 82-10

ontario hydro
research division

To: Dr. Don Mills
Director of Research

VERIFICATION OF VAULT RESPONSE FUNCTION SOLUTION

1. INTRODUCTION

The Canadian concept for spent nuclear fuel disposal consists of enclosing the waste in containers which are then buried deep in sealed vaults in the Canadian shield rock formation. This concept will be modelled mathematically to ensure that the final design is reliable and wll function within the limits set by the regulatory body.

The worst scenario is that the radionuclides may escape the container and travel through the vault into the biosphere. For assessing the concept, it is important to know the time required for the radionuclides to travel through the vault. The model for the mass transport in the vault is based on a set of convection-diffusion differential equations incorporating the radionuclide decay chains (see Appendix I). The solution of this system of partial differential equations is given in [1]. To increase the confidence in the estimates of total radioactive output after burial in the vault, it is necessary to verify the AECL solution independently.

To ensure independent verification, a different solution method was used to calculate the source release rates and response function solutions. Instead of analytically inverting the Laplace transform solution, as AECL had done, the Laplace transform method with a numerical inversion was used.

This report will address the problem and solution methodology in greater detail and discusses the results of our calculations in comparison with those obtained by AECL. The Appendices I, II and III include the mathematical equations and solutions used in the calculations, the computer code used in the solution, and the full set of results obtained by Ontario Hydro, respectively.

2. PROBLEM DESCRIPTION AND VERIFICATION METHODOLOGY

2.1 The Boundary Value Problems

The solutions to the one-dimensional convection-diffusion equations describing mass transport through the finite-length buffer material needed to be verified for two different sets of equations. Each set of equations consisted of a source boundary condition, relating to the container-buffer interface (x=0), an exit boundary condition, relating to the buffer- rock/backfill interface (x=a) and an initial condition (t=0).

The source boundary condition for the first set of equations consisted of unit impulsive input of radionuclides at the container-buffer barrier. The exit boundary condition related the outgoing flux of radionuclide to the concentration at the outer edge through a constant of proportionality called the mass transfer coefficient. The initial conditions were zero. The solution was calculated as the response at the buffer-rock or buffer-backfill boundary (x=a) to a unit impulse of radionuclides at the container-buffer boundary (x=0). These solutions will be referred to as *response functions*.

The second set of equations dealt with the dissolution rate (flux) at the container-buffer boundary (x=0) of the matrix material (made up of radionuclides in a uranium oxide mixture). The source boundary condition was a constant equivalent to the solubility of the matrix material in the container. The exit boundary condition was again given by a mass transfer coefficient at the buffer-rock/backfill boundary. The initial condition was again zero in the buffer. The solution, referred to as *source release functions*, represented the source release rate, in units of flux, of the matrix material.

It may be noted that the source release function is used to calulate the total source release rate of radionuclides, which are dissolved in the matrix. The source release rate of radionuclides, the impulse response functions as described above, and the mean rate of container failure are convoluted over time to give the time-dependent release of radionuclides from the vault.

2.2 The Solution Method

The method chosen to verify the solution to the two problems described above consisted of first solving the equations analytically in Laplace space. Analytical solutions in Laplace space can be derived because the Laplace transformation changed the third order partial differential equations to second-order ordinary

differential equations. When the analytical expression of the Laplace space solutions are obtained, the solution for the response functions and source release functions can be obtained from numericaliy inverted .

An alternate method to verify the AECL solutions was to actually plug in their solutions in the original equations and see if the equations were satisfied. However, this method was not chosen because it would have provided only a global measure of the correctness of the solutions. Moreover, only approximate equality could be proven because even if the solutions satisfied the equations any approximation errors in the calculations would not be identified.

The Laplace inversion approach, however, had the advantage of allowing a point-wise comparison of the results obtained by Laplace inversion and by the AECL method. This local comparison gave stricter measure of correctness than that obtainable by simply plugging in the results. Secondly, the use of the Laplace method obviated the need to have results which were tabulated with respect to variations in space as well as time. If a direct check on the results by plugging in the numerical solutions into the equations was done, then sets of data produced by small variations of the space variable would be needed to do the differentiation numerically. Moreover, the numerical differentiation itself would introduce approximation errors. This would make the accuracy of the test depend on the input data and new sets would need to be produced for more accuracy.

2.3 The Numerical Inversion

The Laplace transforms of the differential equations were solved analytically by the the staff of WNRE, with solutions given in Appendix I. These Laplace-space solutions were then inverted numerically using the Talbot method [2]. The Talbot method was chosen because of the rapid convergence to a desired accuracy and because of fewer restrictions on the kind of singularities that the complex function to be inverted could possess.

The Laplace inversion integral of the Talbot method is easy to compute numerically. Its accuracy can be increased by using a finer partition in the approximation sum. The error in approximation was thus estimated by recalculating the solution for a problem with a partition which was finer by twenty points and then subtracting from this answer the previous solution obtained. The solution itself then consisted of the original Laplace inversion and an estimate of the approximation error of the solution.

The verification of the solutions was implemented using the Fortran programming language on two different computer environments (computer code is listed in the Appendix II). Each of the components of the verification program was tested using sample functions for which analytical solutions were known. The runs were carried

out on both 386/387-based PCs running Microsoft Fortran and a Silicon Graphics 4d/4025 super-mini running Unix with an enhanced Fortran 77 compiler. The program was developed on the PCs and then ported to the super-mini for the complete runs without requiring modification. The PCs calculated the results at rate which would have required 4.2 and 13 days to compute the solution to response function equations and the source release equations, respectively. Instead, the Silicon Graphics computer required only 16 and 50 hours of computation time.

2.4 Standards for Comparison

The solutions to the two problems depended on six parameters for the response functions and seven parameters for the source release functions. Test cases against which solutions were checked, were defined in terms of values for these parameters. Each parameter was allowed to take on three values, one on either end of its range and one in the centre of its range. For the response function equations this gave a total of 3^6 or 729 different sets of parameters and hence 729 different test cases which had to be compared. Similarly, the source release equations required 2187 or 3^7 test cases to be run. The results were checked on a logarithmic time scale. This scale extended from 10^{-3} to 10^9 years in the case of the response functions and 10^{-1} to 10^7 years in the case of the source release equations.

A test statistic was agreed upon to measure the extent of discrepancies between the Ontario Hydro and the AECL solutions. The test statistic was a pointwise test which compared the absolute difference between the two solutions for a given set of parameter values and a given time value. The formula of this test statistic is given in Appendix I. In addition, if both solutions were close enough to zero, then the solutions were assumed to match.

The points were categorized as acceptable, suspicious, or discrepant. The classification is given as follows:

acceptable: if both the solutions were below 10^{-20} or if the test statistic was less than or equal to 2;

suspicious: the points were considered suspicious if at least one of the solutions was greater than 10^{-20} and the test statistic was between 2 and 4;

discrepant: the points were considered discrepant if at least one of the solutions was greater than 10^{-20} and the test statistic was greater than 4.

3. RESULTS

The solutions, for both the response and the source release functions, obtained from AECL and Talbot method were equal to within the numerical approximation error for most of the points. Figures 1 and 2 show the distributions of the number of discrepancies versus the number of cases. It is observed that, for the source release function, 88% of the cases have less than 7 discrepant points out of 73 points calculated, and for the vault response function, 95% of the cases have less than 3 discrepant points out of 109. The average discrepancies per case is 3 points for the source release function, and 1.5 points for the vault response function.

In some cases, there are some oscillations in the Talbot solution at the region where the AECL solution has zero values (see Figure 3). However, the error caused by these oscillations are within the limits of the test statistic. For the cases when discrepancies were found, the discrepant points are located in the non-zero region (Figure 4).

Analysis of the discrepancies mentioned above, by correlating the difference between the two solutions with each of the parameters, produced no conclusive pattern. This means that the difference between the computed values at each time point, which is the numerator of the test statistic, could not be related in a systematic way to the differential equations. Since the denominator of the test statistic depends on the pooled error estimate in the point solutions, a possible explanation for the discrepancy measured by the test statistic is that the error estimates for these points were inaccurate.

Similar to the vault response functions, the Talbot solutions for the source release functions had a large number of discrepancies at the tail end of the function where the concentration started to stabilise (see Figures 5 and 6).

4. CONCLUSION

By comparing the AECL and the Talbot solutions, no perceptible difference between the two were found when plotted on the linear scale. When the logarithmatic scale was used, the difference could be seen. However, no correlation was found between the maximum difference and the test statistic.

In addition, the following conclusions can be drawn:

- In most cases, the maximum discrepancies between the two solutions were several order of magnitude smaller than the maxima of the functions.
- No systematic pattern was found for the discrepancies.

- For the vault response function, most of the test cases have less than 3 discrepant points out of the 109 points calculated.
- For the source release function, most the test cases have less than 7 discrepant points out of 73 points calculated.
- Most of the discrepancies occurred at the area where the responses were stabilised.

Approved:

Dr. A.H. Chung
Manager
Operatons Research Department

Submitted:

A. H. Chan
Mathematician
Statistical Methods & Analysis Unit
Operations Research Department

for Aneel Advani
Summer Student
Statistical Methods & Analysis Unit
Operations Research Department

REFERENCES

[1] Garisto, N.C. and LeNeveu, D.M., "The Vault Model for the Disposal of Used CANDU Fuel: Documentation and Analysis of Scoping Calculations", AECL Research Report No.AECL-9578, Pinawa, July 1989.

[2] Talbot, A., "The Accurate Numerical Inversion of Laplace Transforms", J. Inst. Maths Applics, Vol.23, pp 97-120, 1978.

Figure 1
Number of Discrepancies vs Number of Cases
Vault Response Function

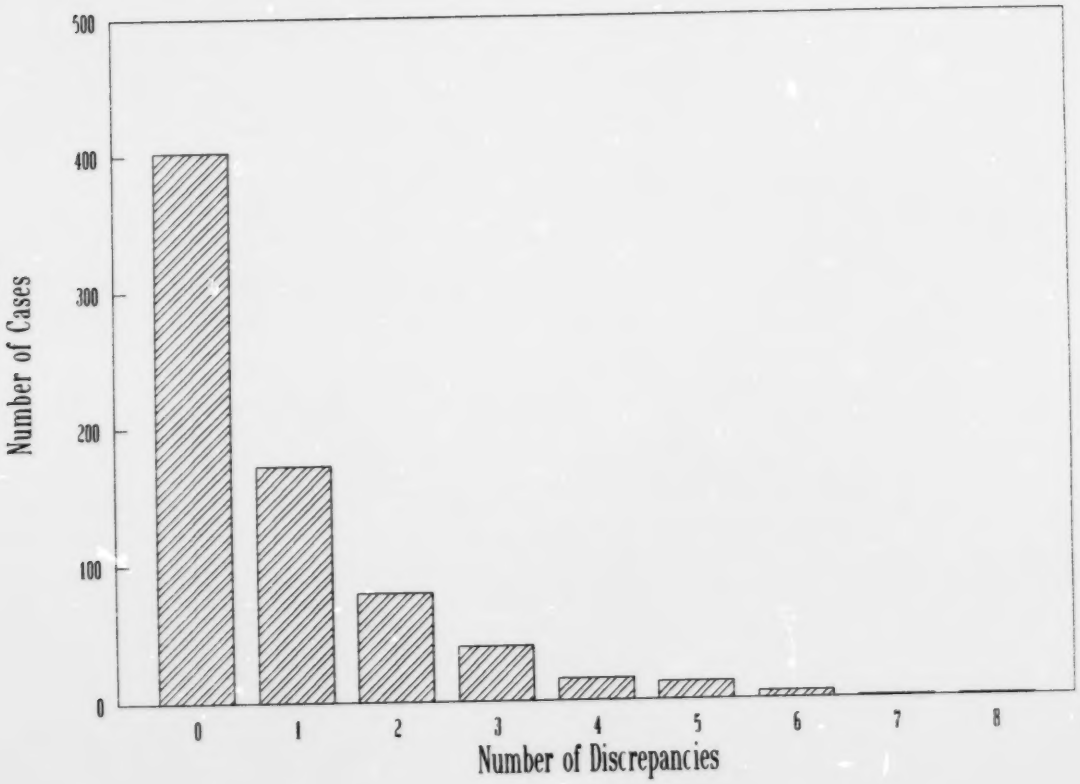


Figure 2
Number of Discrepancies vs Number of Cases
Source Release Function

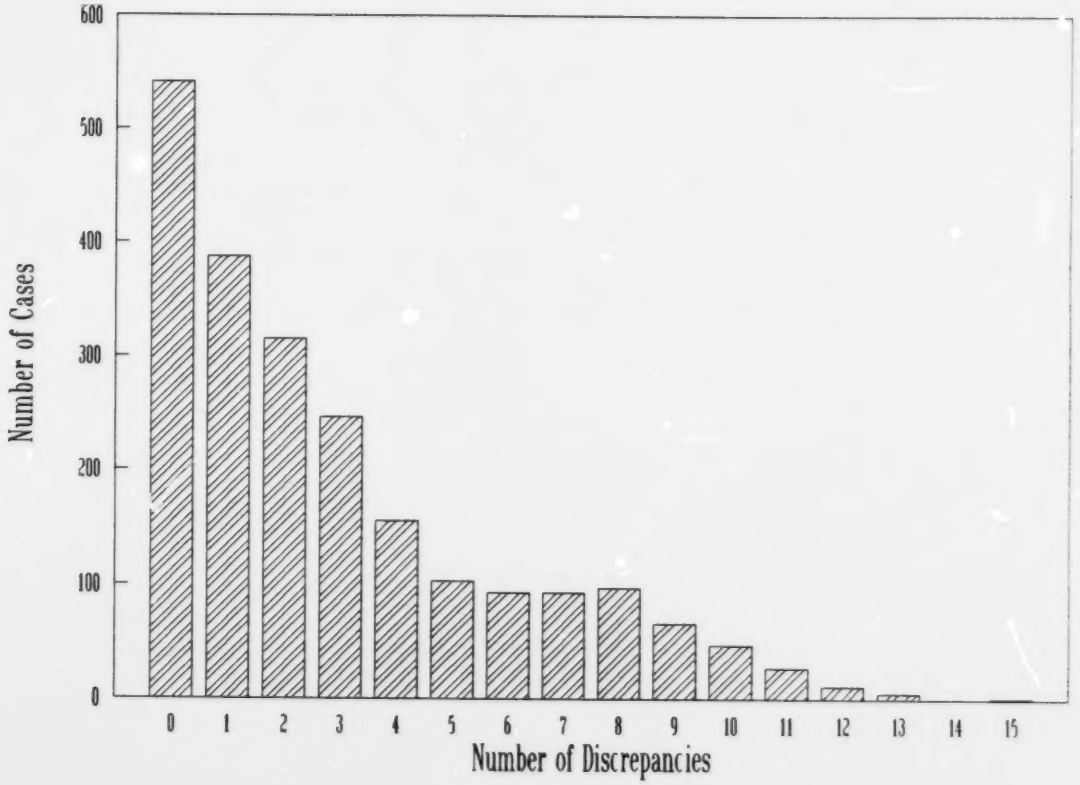
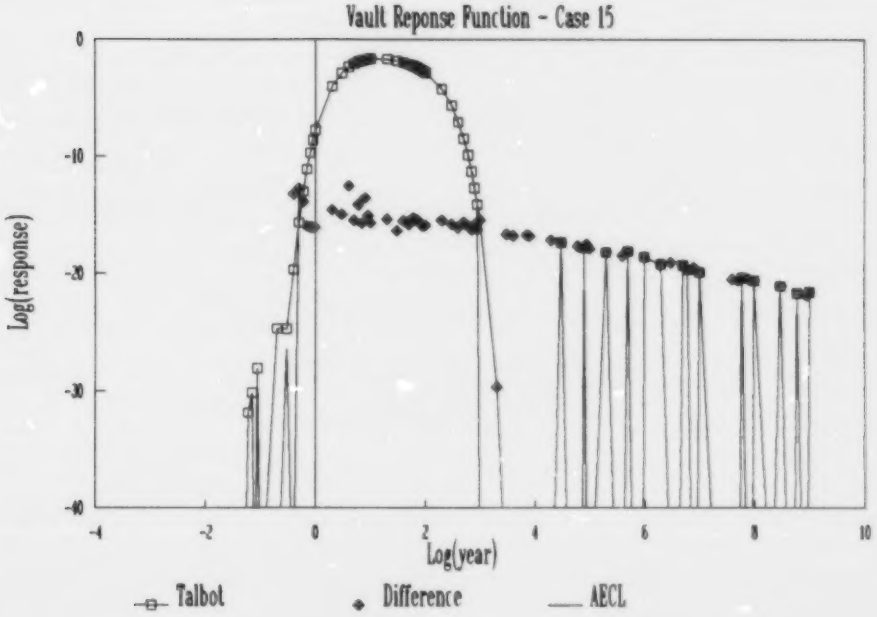


Figure 3
AECL Solution vs Talbot Solution
for the Vault Response Function
Case 15



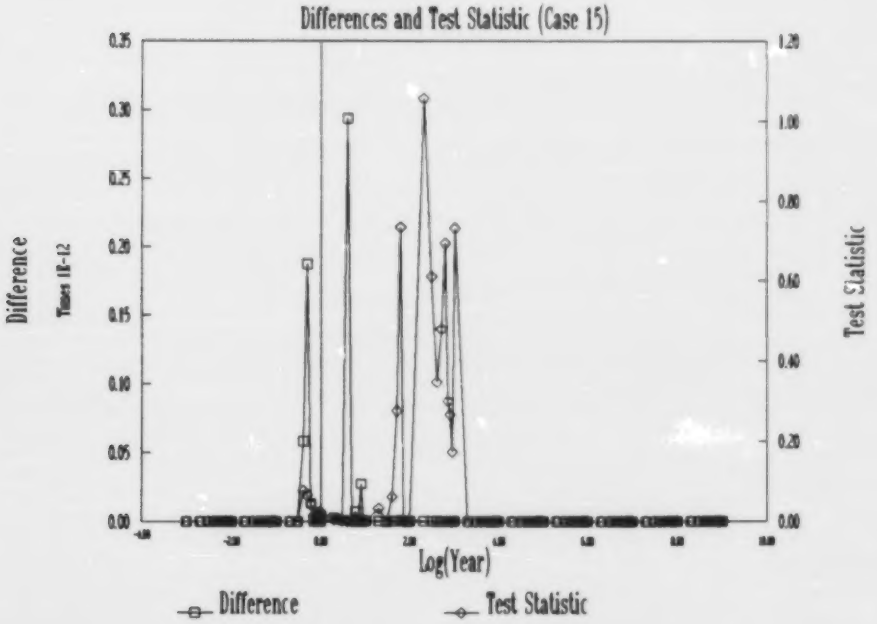
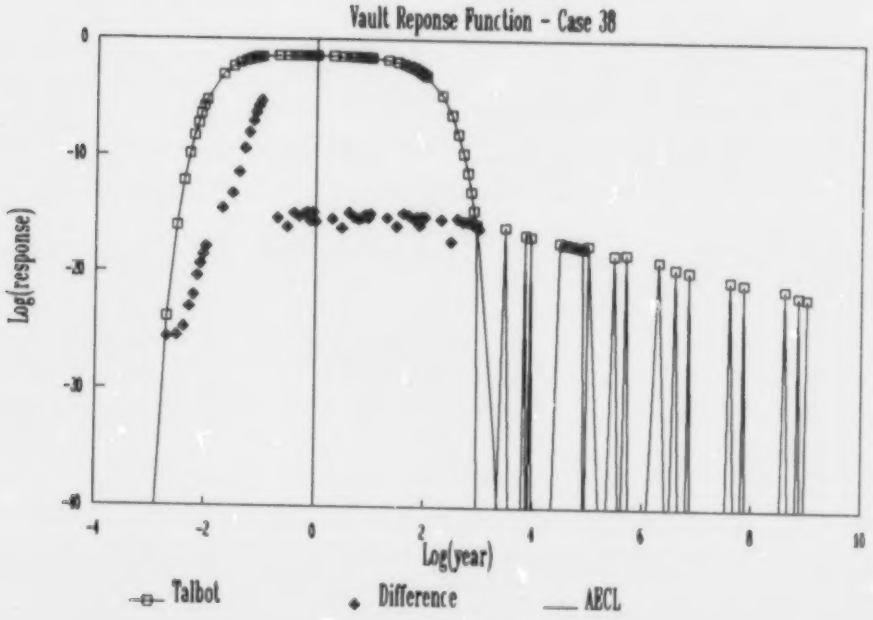
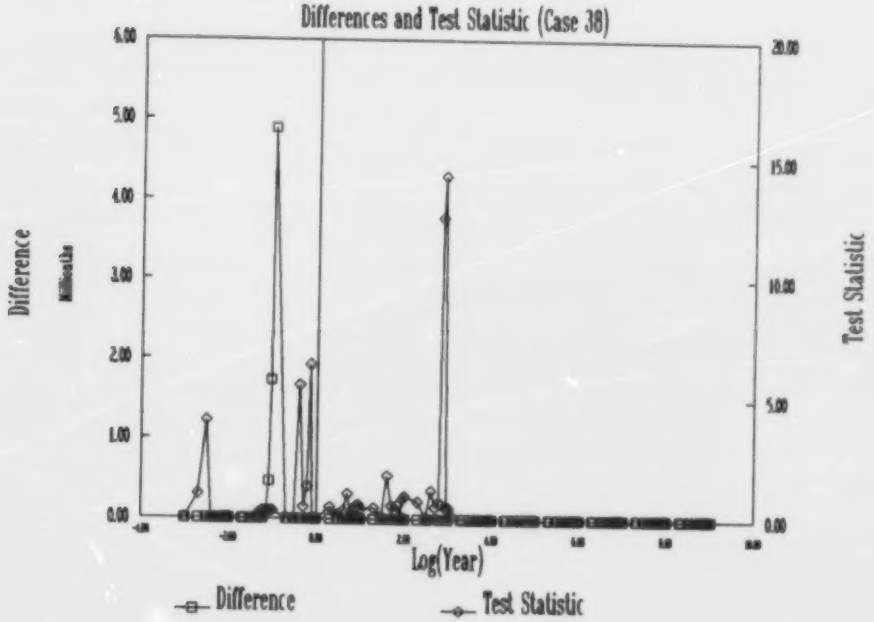
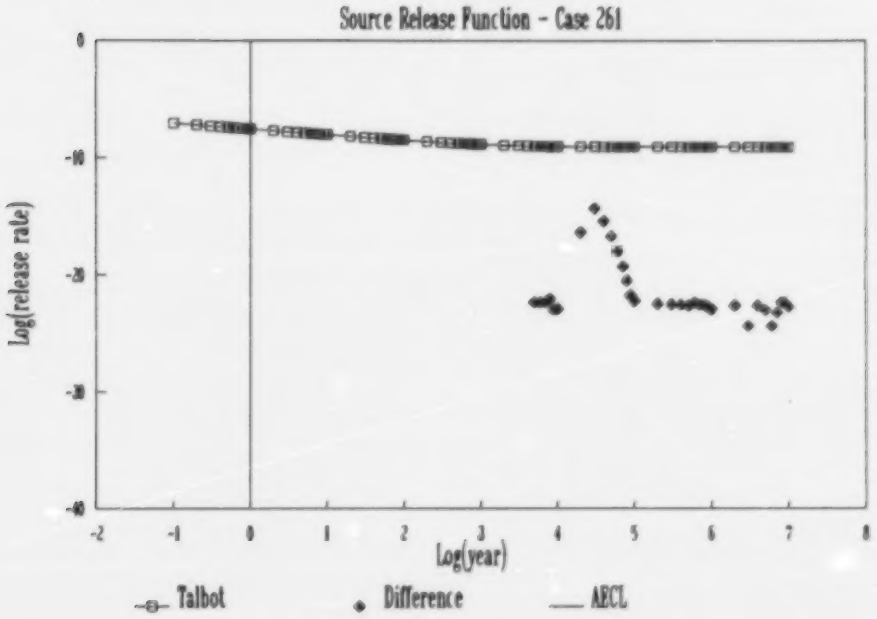


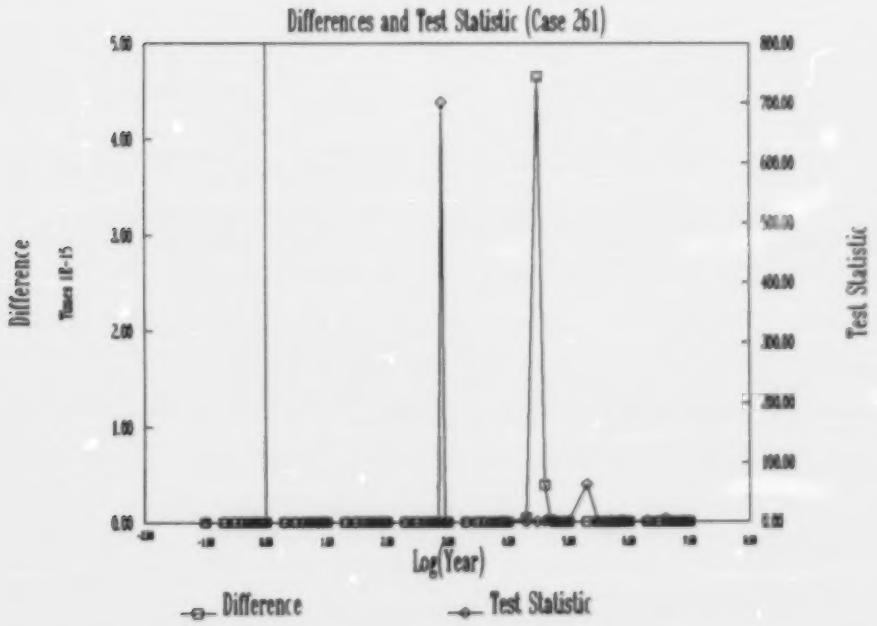
Figure 4
AECL Solution vs Talbot Solution
for the Vault Response Function
Case 38



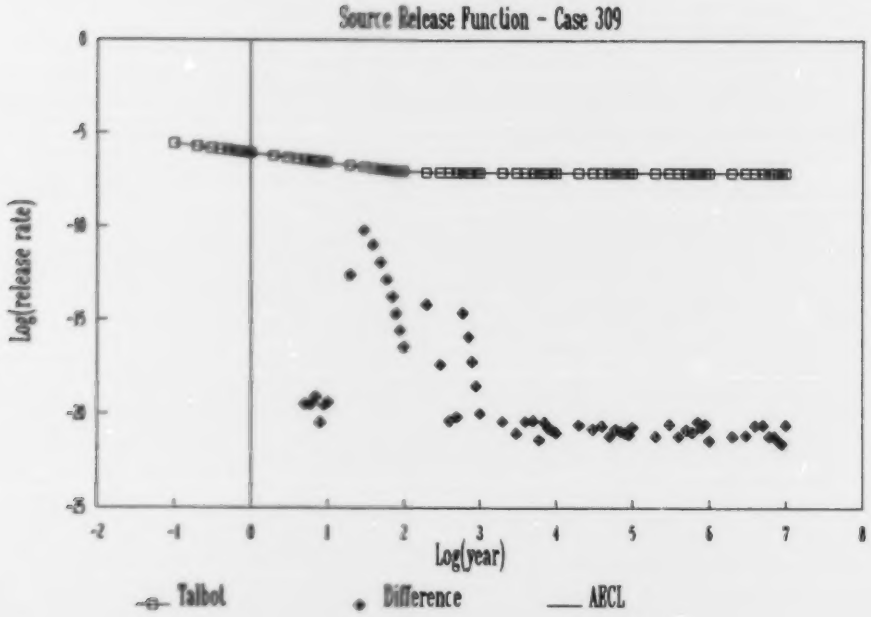


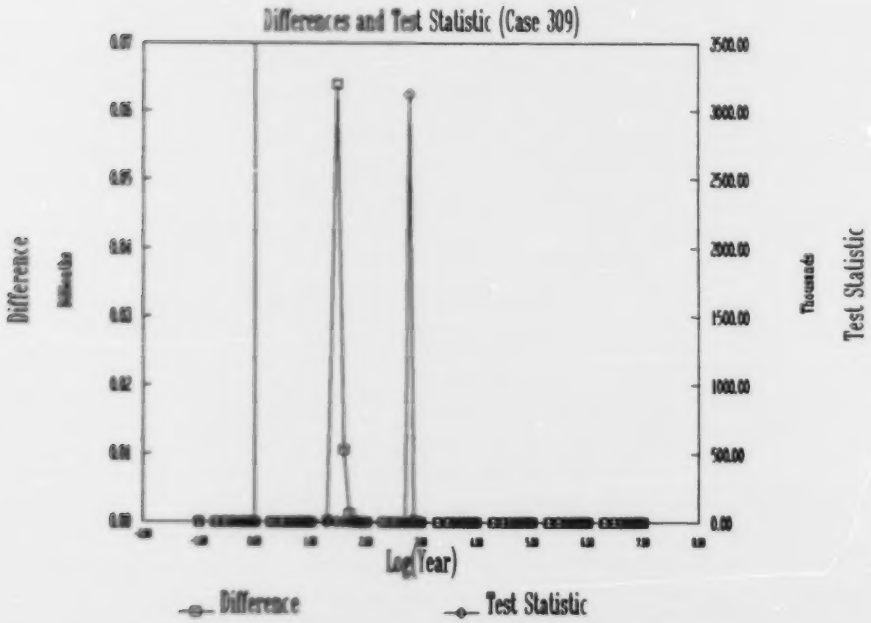
Figures 5
AECL solution vs Talbot Solution
for the Source Release Function
Case 261





Figures 6
AECL Solution vs Talbot Solution
for the Source Release Function
Case 309





APPENDIX I

Mathematical Formulae and Test Parameter Values for Response and Source Release Functions

1. Vault Reponse Function

The following equation defines the ten variables R_{ij}, for each i and j, $1 \leq j \leq i \leq 4$:

$$R_{ij}(x,t) = KC_{ij}(x,t) \qquad [a^{-1}] \text{ at } x = a, \text{ for } 1 \leq j \leq i \leq 4$$

where $C_{ij}(x,t)$ are given by the solutions to the following PDEs:

$$\frac{\partial C_{ij}}{\partial t} = \frac{D}{r_i}\frac{\partial^2 C_{ij}}{\partial x^2} - \frac{V}{r_i}\frac{\partial C_{ij}}{\partial x} - \lambda_i C_{ij}$$

for $[m^{-1}a^{-1}]$ and $1 \leq j = i \leq 4$, and

$$\frac{\partial C_{ij}}{\partial t} = \frac{D}{r_i}\frac{\partial^2 C_{ij}}{\partial x^2} - \frac{V}{r_i}\frac{\partial C_{ij}}{\partial x} - \lambda_i C_{ij} + \frac{\lambda_{i-1} r_{i-1} C_{(i-1)j}}{r_i}$$

for $[m^{-1}a^{-1}]$ and $1 \leq j < i \leq 4$, with initial conditions,

$$C_{ij}(x,t) = 0 \qquad [m^{-1}] \text{ at } t = 0 \text{ for } 0 \leq x \leq a$$

and boundary conditions

$$-D\frac{\partial C_{ij}}{\partial x} + VC_{ij} = I_j^u \delta(t)\delta_{ij}$$

$[a^{-1}]$ at $x = 0$

where $\delta(t)$ is the Dirac delta function, δ_{ij} is the Kronecker delta function , and

$$-D\frac{\partial C_{ij}}{\partial x} + VC_{ij} = KC_{ij}$$

$[a^{-1}]$ at $x = a$

Applicable ranges of the parameters are, fo r the nuclide i:

0.25	$\leq$	a	≤ 1.4	medium thickness [m]
9×10^{-7}	$\leq$	D	$\leq 2.7\times10^{-2}$	diffusion coefficient $[m^2/a]$
0.0	$\leq$	K	$\leq \max\{50D/a, V/2\}$	mass transfer coefficient [m/a]
10^{-4}	$\leq$	r_i	$\leq 2\times10^4$	capacity factor
0.0	$\leq$	t	$\leq 10^7$	time [a]

$0.0 \leq V \leq 10D/a$		Darcy velocity [m/a]
$I_i^u = 1.0$		unit initial amount
$0.0 \leq \lambda \leq 5.41 \times 10^{-2}$		decay constant [a^{-1}]
C_{ij}		concentration [m^{-1}]

2. Source Release Function

The detailed derivation of the source release function is given in [1]. The following equation calculates the variable:

$$F(x,t) = A\left[- D\frac{\partial C}{\partial x} + VC\right]$$

[mol/a] at $x = 0$

where C is the solution to the following partial differential equation:

$$\frac{\partial C}{\partial t} = \frac{D}{r}\frac{\partial^2 C}{\partial x^2} - \frac{V}{r}\frac{\partial C}{\partial x} - \lambda C$$

[mol $m^{-3}a^{-1}$]

with initial condition

$C(x,t) = 0$ [mol/m^3] for $0 \leq x \leq a$ and $t = 0$

and boundary conditions

$C(x,t) = C^o$ [mol/m^3] at $x = 0$ and $t > 0$

and

$$- D\frac{\partial C}{\partial x} + VC = KC$$

[mol $m^{-2}a^{-1}$] at $x = a$

Applicable ranges of the parameters are:

$0.25 \leq a \leq 1.4$		medium thickness [m]
$2 \times 10^6 \leq A \leq 4 \times 10^6$		area [m^2]
$10^{-10} \leq C^o \leq 10^{-1}$		solubility [mol/m^3]
$3.6 \times 10^{-7} \leq D \leq 3.6 \times 10^{-4}$		diffusion coefficient [m^2/a]
$0.0 \leq K \leq \max\{50D/a, V/2\}$		mass transfer coefficient [m/a]
$1.67 \times 10^{-1} \leq r \leq 1.67 \times 10^4$		capacity factor
$0.0 \leq t \leq 10^7$		time [a]
$0.0 \leq V \leq 10D/a$		Darcy velocity [m/a]
$0.0 \leq \lambda \leq 1.55 \times 10^{-10}$		decay constant [a^{-1}]

2. Solutions in Laplace Space

The solutions for the differenece equations in the Laplace space were provided by the WNRE.

2.1 Solution for the Vault Response Function

The solution in the Laplace space for a single member response function for semi-infinite flow segment is:

$$r_j(a,s) = \exp\left\{\frac{-2\Phi_j a}{V + \beta_j}\right\} \qquad 1 \le j \le n, [\,],$$

where

$$\Phi_j = r_j (s + \lambda_j)$$

and

$$\beta_j = \sqrt{V^2 + 4D\Phi_j} \qquad 1 \le j \le n, \text{ [length/time]}.$$

For a finite flow segment with a mass transfer boundary coefficient, the solution in the Laplace space is:

$$r_j(a,s) = \frac{2K\beta_j}{(2D\Phi_j + KV)\tanh(a\beta_j/2D) + K\beta_j} \cdot \frac{\exp\left\{\frac{-2\Phi_j a}{V + \beta_j}\right\}}{1 + \exp\left\{\frac{-a\beta_j}{D}\right\}}$$

2.2 Solution for the Source Release Functions

The solution for the source release function is:

$$F(p) = \frac{sC^oA}{p} + \frac{DC^oqA}{p}\left[\frac{Dq\sinh(qa) + (K-s)\cosh(qa)}{Dq\cosh(qa) + (K-s)\sinh(qa)}\right]$$

where $s = V/2$, $\beta = -V^2/4rD$, $q = \{r(p + \lambda - \beta)/D\}^{1/2}$, V is the Darcy velocity, D is the diffusion constant, λ is the decay constant, K is the mass transfer coefficient, r is the

capacity factor, C^o is the solubility, a is the thickness of the medium, A is the area, and p is the Laplace transform variable.

3. The Parameter Values for the Test Cases

For each parameter, 3 values (low, median, upper) were chosen to test the program. The chosen values are given in the following tables:

Table 1
Test Case Values For the Finite-interval Response Functions

Parameter	Sym	Lower Value	Median	Upper Value
Decay Const.	λ	0.0	3.25×10^{-6}	5.41×10^{-2}
Capacity Factor	r	10^{-4}	1.414	2×10^{4}
Diffusion Cost.	D	9×10^{-7}	1.56×10^{-4}	2.7×10^{-2}
Thickness	a	2.5×10^{-1}	8.25×10^{-1}	1.40
Darcy Velocity	V	0.0	$\min\{10^{-7}, 10D/a\}$	$\min\{10^{-4}, 10D/a\}$
Mass Transf Coef	K	$\max\{V/2, \min\{10^{-7}, 50D/a\}\}$	$\max\{V2, \min\{3.16 \times 10^{-6}, 50D/a\}\}$	$\max\{V/2, 50D/a\}$

Table 2
Test Case Values For the Source Release Functions

Parameter	Sym	Lower Value	Median	Upper Value
Area*Solubility	AC^o	2×10^{-4}	9.48	4×10^{5}
Decay Const.	λ	0.0	3.25×10^{-6}	5.41×10^{-2}
Capacity Factor	r	1.67×10^{-1}	52.81	1.67×10^{4}
Diffusion Cost.	D	3.6×10^{-7}	1.14×10^{-5}	3.6×10^{-4}
Thickness	a	2.5×10^{-1}	8.25×10^{-1}	1.40
Darcy Velocity	V	0.0	$\min\{10^{-7}, 10D/a\}$	$\min\{10^{-4}, 10D/a\}$
Mass Transf Coef	K	$\max\{V/2, \min\{10^{-7}, 50D/a\}\}$	$\max\{V/2, \min\{3.16 \times 10^{-6}, 50D/a\}\}$	$\max\{V/2, 50D/a\}$

5. Formula for the Test Statistic

The test statistic T for a given time t is defined by

$$T = \frac{\left| Z_1(t) - Z_2(t) \right|}{\sqrt{\varepsilon_1^2(t) + \varepsilon_2^2(t)}}$$

where $Z_1(t)$ = the value computed from the AECL's algorithm at time t
$Z_2(t)$ = the value computed from the Talbot's algorithm at time t
$\varepsilon_1(t)$ = the computed error of the AECL solution at time t.
$\varepsilon_2(t)$ = the computed error of the Talbot solution at time t.

APPENDIX II

Computer Code

```
C
C ***** vf
C
      PROGRAM VERIFICATION
C
      EXTERNAL FS,TALBOT,AVG,SD,MAXV,WRTEXT
      DOUBLE PRECISION AVG,SD,MAXV
      COMPLEX*16 FS
      INTEGER J,I,NCHAIN,RUNNO,NMAX,N,I1,NITER,NITER2
      INTEGER FN,M,ISEQ,ISEQ2,NI,RU,ZN,ZP,TF,TE,TZ
      PARAMETER (NCHAIN=4,NMAX=120)
      COMPLEX*16 RR(NCHAIN,NCHAIN)
      CHARACTER*80 SKIPL
      CHARACTER*12 FILEN1,FILEN2
      CHARACTER EXTN*3
      CHARACTER*12 FILEN3,FILEN4
      DOUBLE PRECISION L,K,D,A,V,KAPPA,ALPHA,C0,AR,PEAK(NMAX),
     1                 RSPINS(NMAX),LGRSPINS(NMAX),ERRINS(NMAX),
     1                 TIME(NMAX),FINV(NMAX),LGFINV(NMAX),FERR(NMAX),
     1                 RELERR(NMAX),LGT(NMAX),Z(NMAX),Z3(NMAX),
     1                 TEST1,TEST2,TEST3,R1,R2,R3,LIMIT,FINV2(NMAX),
     1                 MERR,MFUNC,AERR,MZ,AZ
C
      OPEN(UNIT=9,FILE='ftoutpt3.dat')
      REWIND(9)
      OPEN(UNIT=14,FILE='ftoutpt4.dat')
      REWIND(14)
      OPEN(UNIT=16,FILE='finparam.set',STATUS='OLD')
      REWIND(16)
      READ(16,*) J
      READ(16,*) I
      READ(16,*) ALPHA
      READ(16,*) NITER
      READ(16,*) NITER2
      READ(16,*) FN
      READ(16,*) LIMIT
      READ(16,*) NI
      do 7777 ru=1,ni
      READ(16,*) ISEQ
      IF (FN .EQ. 2) THEN
       CALL WRTEXT(EXTN,ISEQ)
       FILEN3='insres.'//EXTN
       filen2='ftoutput.'//extn
      ENDIF
      IF (FN .EQ. 3) THEN
       IF (ISEQ .LT. 1000) THEN
        CALL WRTEXT(EXTN,ISEQ)
        FILEN3='flomat0.'//EXTN
        FILEN2='ftoutpt0.'//extn
       ENDIF
```

```
        IF (ISEQ .GT. 1999) THEN
          ISEQ2 = ISEQ-2000
          CALL WRTEXT(EXTN,ISEQ2)
          FILEN3='flomat2.'//EXTN
          filen2='ftoutpt2.'//extn
        ELSEIF (ISEQ .GT. 999) THEN
          ISEQ2 = ISEQ-1000
          CALL WRTEXT(EXTN,ISEQ2)
          FILEN3='flomat1.'//EXTN
          filen2='ftoutpt1.'//extn
        ENDIF
       ENDIF
       FILEN4='ftout2.'//EXTN
C      WRITE(*,*) ' The filename is ',filen3
       OPEN(UNIT=7,FILE=FILEN3,STATUS='OLD')
       REWIND(7)
       OPEN(UNIT=8,FILE='finparam.dat',STATUS='OLD')
       REWIND(8)
C
C     Read from aecltemp and write header into finparam.dat
C
 45    FORMAT(A72)
       READ(7,45) SKIPL
       IF (FN .EQ. 2) READ(7,'(26X,I3)') RUNNO
       IF (FN .EQ. 3) READ(7,'(26X,I5)') RUNNO
       READ(7,45) SKIPL
       READ(7,45) SKIPL
       READ(7,45) SKIPL
       IF (FN .EQ. 2) THEN
       READ(7,'(2(1X,I3,3X),2(1X,1PD15.8,1X))') J,I,L,K
       ENDIF
       IF (FN .EQ. 3) THEN
       READ(7,'(4(1X,1PD12.4,1X))') C0,AR,L,K
       ENDIF
       READ(7,45) SKIPL
       READ(7,'(4(1X,1PD12.4,1X))') D,A,V,KAPPA
       READ(7,45) SKIPL
       READ(7,45) SKIPL
       READ(7,45) SKIPL
       READ(7,45) SKIPL
       IF (FN .EQ. 2) WRITE(8,'(I3)') RUNNO
       IF (FN .EQ. 3) WRITE(8,'(I5)') RUNNO
       WRITE(8,'(I1)') J
       WRITE(8,'(I1)') I
       WRITE(8,'(1PD10.4)') ALPHA
       WRITE(8,'(I3)') NITER
       WRITE(8,'(I3)') NITER2
       WRITE(8,'(I1)') FN
 46    FORMAT(1PD12.4)
       WRITE(8,46) V
       WRITE(8,46) D
       WRITE(8,46) KAPPA
       WRITE(8,46) A
 47    FORMAT(1PD15.8)
       WRITE(8,47) K
       WRITE(8,47) K
       WRITE(8,47) K
       WRITE(8,47) K
```

```
      WRITE(8,47) L
      WRITE(8,47) L
      WRITE(8,47) L
      WRITE(8,47) L
      R1 = C0*AR
      IF (FN .EQ. 3) WRITE(8,47) R1
C
C     Read from aecltemp into arrays for ressiil to work on
C
 90   FORMAT(1X,1PD8.2,1X,1PD24.16,2X,1PD8.2)
 91   FORMAT(1X,1PD8.2,1X,2(1PD7.1,2X))
 92   FORMAT(1X,1PD8.2,1X,2(1PD24.16,2X))
      N=1
 101  READ(7,*, END=130) R1,R2,R3
      BACKSPACE(7)
      IF (FN .EQ. 3) READ(7,92) TIME(N),RSPINS(N),ERRINS(N)
      IF (FN .EQ. 2) THEN
       IF (DABS(R2) .GT. 0.0D0) THEN
         READ(7,90) TIME(N),RSPINS(N),ERRINS(N)
       ELSE
         READ(7,91) TIME(N),RSPINS(N),ERRINS(N)
       ENDIF
      ENDIF
      N = N + 1
      GOTO 101
 130  CONTINUE
      N = N - 1
      CLOSE(7,STATUS='KEEP')
C
C           LAPLACE INVERSION AND INTEGRATION LOOP
C
 70   FORMAT(A21,I3,1X,1PE8.2,1X,1PE10.3,1X,1PE10.3)
C
      DO 140 I1=1,N
        CALL TALBOT(FINV(I1),TIME(I1),NITER,ALPHA,J,I,RR,NCHAIN)
        IF (DABS(FINV(I1)).LT.10D-38) FINV(I1)=0.0D0
        CALL TALBOT(FINV2(I1),TIME(I1),NITER2,ALPHA,J,I,RR,NCHAIN)
        IF (DABS(FINV2(I1)).LT.10D-38) FINV2(I1)=0.0D0
        RELERR(I1)=dabs(FINV(I1)-RSPINS(I1))
        FERR(I1) = DABS(FINV(I1)-FINV2(I1))
 140  CONTINUE
C
      CLOSE(8,STATUS='KEEP')
      OPEN(UNIT=7,FILE=FILEN2)
      REWIND(7)
 80   FORMAT(1X,1PE8.2,2(1X,1PD24.16,2X,1PE8.2))
 81   FORMAT(1X,1PE9.2,3X,1PE10.3,3X,1PE10.3,3X,1PE10.3,3X,1PE10.3)
C
      DO 145 I1=1,N
        LGT(I1) = DLOG10(TIME(I1))
 145  CONTINUE
C
      ZN = 0
      ZP = 0
C
      DO 146 I1=1,N
C
      IF (FINV(I1) .GT. LIMIT) THEN
```

```
        LGFINV(I1) = DLOG10(FINV(I1))
      ELSE
        LGFINV(I1) = DLOG10(LIMIT)
      ENDIF
C
      IF (RSPINS(I1) .GT. LIMIT) THEN
        LGRSPINS(I1) = DLOG10(RSPINS(I1))
      ELSE
        LGRSPINS(I1) = DLOG10(LIMIT)
      ENDIF
C
      IF ((ERRINS(I1) .EQ. 0.0D0) .AND. (FERR(I1) .EQ . 0.0D0)) THEN
        Z(I1) = 0.00D0
      ELSE
        Z(I1) = DABS(RELERR(I1)/DSQRT((FERR(I1)**2)+(ERRINS(I1)**2)))
      ENDIF
C
      IF ((FINV(I1) .LE. LIMIT) .AND. (RSPINS(I1) .LE. LIMIT)) THEN
        Z3(I1) = 0.00D0
      ELSE
        Z3(I1) = -1.0D0
        ZN = ZN + 1
C
        IF ((Z(I1) .LE. 4.0D0) .AND. (Z(I1) .GT. 2.0D0)) THEN
          Z3(I1) = 1.0D0
          ZN = ZN - 1
          ZP = ZP + 1
        ENDIF
C
        IF (Z(I1) .LE. 2.0D0) THEN
          Z3(I1) = 0.0D0
          ZN = ZN - 1
        ENDIF
C
      ENDIF
C
 146  CONTINUE
C
      DO 150 I1=1,N
      WRITE(7,81) LGT(I1),FINV(I1),relerr(I1),RSPINS(I1),Z(I1)
 150  CONTINUE
C
C     Calculate various test statistics
C
      M = 0
      DO 160 I1=1,N
        PEAK(I1)=0.0D0
        IF ( DABS(MAX(FINV(I1),RSPINS(I1))) .GT. 10.0D-20 ) THEN
          PEAK(I1) = 1.0D0
          M = M + 1
        ENDIF
 160  CONTINUE
      IF (M .GT. 1) THEN
        TEST1=AVG(PEAK,M,RELERR,N)
        TEST2=SD(PEAK,M,RELERR,N,TEST1)
      ELSE
        TEST1 = 0.0D0
        TEST2 = 0.0D0
```

```
      ENDIF
      IF (TEST2 .EQ. 0.0D0) THEN
        TEST3 = 0.0D0
      ELSE
        TEST3=TEST1*DSQRT(DBLE(M))/TEST2
      ENDIF
 77   FORMAT(52X,1PE10.3)
 78   FORMAT(52X,I3)
      WRITE(7,*) '    '
C
      CLOSE(7,STATUS='KEEP')
      MZ=MAXV(Z,Z,N,TZ)
      AZ=AVG(PEAK,M,Z,N)
      MERR=MAXV(RELERR,RELERR,N,TE)
      AERR=AVG(PEAK,M,RELERR,N)
      MFUNC=MAXV(FINV,RSPINS,N,TF)
 59   FORMAT(2(1X,I5),1X,1PE10.3,1X,3(1PE10.3,2X),2(1X,I3))
      WRITE(9,59) RUNNO,M,AERR,MERR,LGT(TE),MFUNC,ZP,ZN
      WRITE(14,59) RUNNO,M,AZ,MZ,LGT(TZ),MFUNC,ZP,ZN
 7777 continue
      CLOSE(16,STATUS='KEEP')
      CLOSE(14,STATUS='KEEP')
      CLOSE(9,STATUS='KEEP')
      STOP 'ENDED OK'
      END
C
C     Some Functions     ***********************************************
C
      DOUBLE PRECISION FUNCTION MAXV(A1,A2,N,T)
C
      INTEGER N,I,T
      DOUBLE PRECISION A1(N),A2(N),R2
C
      MAXV=MAX(A1(1),A2(1))
      DO 100 I=1,N
        R2 = MAX(A1(I),A2(I))
        IF (R2 .GT. MAXV) THEN
          MAXV=R2
          T = I
        ENDIF
 100  CONTINUE
C
      RETURN
      END
C
C
      DOUBLE PRECISION FUNCTION AVG(PEAK,M,LIST,N)
C
      INTEGER M,N,I
      DOUBLE PRECISION PEAK(N),LIST(N),SUM
C
      SUM = 0.0D0
      DO 100 I=1,N
      SUM = SUM + PEAK(I)*LIST(I)
 100  CONTINUE
      IF (M .GT. 0) THEN
        AVG = SUM/DBLE(M)
      ELSE
```

```
        AVG = 0.0D0
      ENDIF
C
      RETURN
      END
C
C
C
      DOUBLE PRECISION FUNCTION SD(PEAK,M,LIST,N,AVG)
C
      INTEGER M,N,I
      DOUBLE PRECISION PEAK(N),LIST(N),AVG,SUM,A2
C
      SUM = 0.0D0
      A2 = AVG
      DO 100 I=1,N
        IF (PEAK(I) .GT. 0.0D0) THEN
          SUM = SUM + (LIST(I)-A2)*(LIST(I)-A2)
        ENDIF
 100  CONTINUE
        SD = DSQRT((SUM)/DBLE(M-1))
C
      RETURN
      END
C
C
C
      SUBROUTINE WRTEXT(EXTN,ISEQ)
C
      CHARACTER EXTN*3
      CHARACTER FRMT*4
      INTEGER NDG,ISEQ
C
      EXTN = '   '
      FRMT = '(i )'
      IF (ISEQ .EQ. 0) THEN
        EXTN = '000'
      ELSE
        NDG = INT(LOG10(FLOAT(ISEQ))) + 1
        WRITE(FRMT(3:3), 10) NDG
        WRITE(EXTN(1:NDG),FRMT) ISEQ
      ENDIF
 10   FORMAT(i1)
      RETURN
      END
C
C
C             Talbot Subroutine
C
C
      SUBROUTINE TALBOT(FT,T,N,SIGMA,J,I,RR,NC)
C
C
C     The Talbot subroutine inverts a Laplace Transform which has all
C     its singularities on the real axis.  It was written by John Barker
C     10/7/1980, W. Heinrich 5/9/1985, and D. Leneveu 16/11/1987
C
C
```

```
C     ***************************************************
C     FT    is the required inverse transform at time t
C     FS    is the transform requiring inversion
C     FS(S) is a complex*16 function on an arbitrary complex argment S
C     N     is the number of points to be taken in the quadrature along
C           the  Green's contour
C     SIGMA is the shift parameter used to ensure the integration
C           contour is to the right of all the singularities of FS
C     ***************************************************
C
      DOUBLE PRECISION NU,LAMBDA
      INTEGER N,NM1,K,KA,J,I,NC
      COMPLEX*16 S(1000),DS(1000),ZZ,FS,SUM,B,B1,B2,RR(NC,NC)
      DOUBLE PRECISION Z,PI,TAU,PIBYN,THETA,ALPHA,PSI,CP
      DOUBLE PRECISION SP,V2,FT,T,SIGMA
      DATA Z/0.0D0/,PI/3.14159265358979932D0/,NU/1.0D0/,
     +TAU/6.0D0/
C
      PIBYN = PI/DFLOAT(N)
      ZZ    = DCMPLX(Z,Z)
      LAMBDA= TAU/T
      NM1   = N-1
C
      DO 10 K = 1,NM1
      THETA   = DFLOAT(K)*PIBYN
      ALPHA   = THETA/DTAN(THETA)
      S(K)    = DCMPLX(ALPHA,THETA)
      DS(K)   = DCMPLX(NU,THETA+ALPHA*(ALPHA-1.0D0)/THETA)*.5D0
 10   CONTINUE
C
      PSI   = TAU*PIBYN
      CP    = 2.0D0*DCOS(PSI)
      SP    = DSIN(PSI)
      B     = ZZ
      B1    = B
 15   CONTINUE
           DO 20 KA = 1 , NM1
           K     = N-KA
           V2    = DEXP(DMAX1(TAU*DREAL(S(K)),-160.0D0))
           B2    = B1
           B1    = B
           B     = CP*B1-B2+V2*DS(K)*FS(LAMBDA*S(K)+SIGMA,J,I,RR,NC)
 20   CONTINUE
C
      SUM = DEXP(TAU)*FS(LAMBDA+ZZ+SIGMA,J,I,RR,NC)*.5D0+
     +      CP*B-2.0D0*(B1-B*DCMPLX(Z,SP))
           FT= LAMBDA*DREAL(SUM)/DFLOAT(N)*EXP(SIGMA*T)
 25   CONTINUE
C
      RETURN
      END
C
C     ****  Functins FS  ****
C
C     ***************************************************************
C      VF HAS SAME DEFN AND ALL FINPARAM.DAT VAR ARE INCLUDED;
C      GET RID OF EXCESS PARAMS IN TALBOT,FS AND FINPARAM.DAT.
C      CAN ALSO USE ENTRY STATEMENTS INSTEAD OF FN .EQ. ...
```

```
C     ****************************************************************
C
      INTEGER J,I,N
      COMPLEX*16 S,RR(N,N)
C
      COMPLEX*16 HYPTAN
      EXTERNAL HYPTAN
C
      CHARACTER*80 S1
      INTEGER I1,I2,JTEMP,ITEMP
      DOUBLE PRECISION RL1
      COMPLEX*16 C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12,C13,C14
      COMPLEX*16 CX(30)
C
      INTEGER FN
      COMPLEX*16 P(4),B(4),Q(4),SANDB(4)
      DOUBLE PRECISION K(4),L(4),V,D,A,KAPPA,C0,X,T
C
      PHI(J,S) = (L(J)+S)*K(J)
      BEE(J,S) = CDSQRT((V*V)+(4.0D0*D*PHI(J,S)))
C
      R(J,I)=(L(I-1)*K(I-1)*RR(J,I-1)-L(J)*K(J)*RR(J+1,I))/(P(I)-P(J))
C
C
      REWIND(8)
C
C     SKIP FIRST SIX ARGS
C
      READ (8,*) I2
      READ (8,*) I2
      READ (8,*) I2
      READ (8,*) RL1
      READ (8,*) I2
      READ (8,*) I2
C
C     GET CHOICE OF SI OR FI CALCULATION
C
      READ (8,'(I3)') FN
C
      READ (8,*) V,D,KAPPA,A
      READ (8,*) K(1),K(2),K(3),K(4),L(1),L(2),L(3),L(4)
      IF (FN .EQ. 3) READ (8,*) C0
C
      JTEMP = J
      ITEMP = I
C
      DO 109, J=1,N
       P(J) = (L(J)+S)*K(J)
       Q(J) = CDSQRT(P(J)/D)
       SANDB(J) = S-((V*V)/(4*D*K(J)))
 109  CONTINUE
C
      DO 108, J=1,N
       B(J) = CDSQRT((V*V)+(4.0D0*D*P(J)))
 108  CONTINUE
C
      DO 103, I=1,N
         DO 102, J=1,N
```

```
            IF (I.LT.J) RR(J,I) = DCMPLX(0.0,0.0)
 102      CONTINUE
 103   CONTINUE
C
      DO 105, J=1,N
C
C  1=SEMI-INFINITE, 9 = SAMPLE, 2 = FINITE RESP, 3 = SOURCE RELEASE
C
      IF (9 .EQ. FN) GOTO 444
      IF (1 .EQ. FN) GOTO 222
      IF (2 .EQ. FN) GOTO 222
      IF (3 .EQ. FN) GOTO 555
      GOTO 333
C
C     SOURCE RELEASE CALCULATIONS
C
 555  CONTINUE
      CX(1) = D*Q(J)
      CX(2) = KAPPA-(V/2)
      IF (SANDB(J) .EQ. DCMPLX(0.0D0,0.0D0)) STOP ' S+BETA = 0'
      CX(3) = V*C0/(2*SANDB(J))
      IF (Q(J) .NE. DCMPLX(0.0D0,0.0D0)) THEN
       CX(4) = CX(1)*C0/SANDB(J)
       CX(5) = A*Q(J)
       CX(6) = HYPTAN(CX(5))
       CX(7) = CX(2) + (CX(1)*CX(6))
       CX(8) = CX(1) + (CX(2)*CX(6))
       CX(9) = CX(7)/CX(8)
       CX(10) = CX(3)+(CX(4)*CX(9))
      ELSE
       CX(10) = CX(3)
      ENDIF
C
      RR(J,J) = CX(10)
      GOTO 333
C
C
C
 222  CONTINUE
C
C          RR(J,J) = CDEXP((-2.0D0*P(J)*A)/(V+B(J)))
C
         RR(J,J) = DCMPLX(0.0D0,0.0D0)
         CX(20) = -2.0D0*P(J)*A
         CX(21) = V+B(J)
         IF (CDABS(CX(20)).LT.(704*CDABS(CX(21)))) THEN
           RR(J,J) = CDEXP(CX(20)/CX(21))
         ENDIF
C
      IF (1 .EQ. FN) GOTO 333
C
C          RR(J,J) = FINITE RESPONSE FUNCTION
C
         CX(1) = RR(J,J)
         CX(6) = A*B(J)/D
         CX(2) = CX(1)
         IF (CDABS(CX(6)).LT.704) THEN
            CX(2)=CX(1)/(1.0D0+CDEXP(-1.0D0*CX(6)))
```

```
         ENDIF
         CX(3) = KAPPA*B(J)
         CX(4) = (KAPPA*V)+(2.0D0*D*P(J))
         CX(5) = CX(6)/2.0D0
         CX(8) = HYPTAN(CX(5))
         CX(9) = (CX(4)*CX(8))+CX(3)
         CX(10) = 2.0D0*CX(3)
         CX(11) = CX(10)/CX(9)
         CX(12) = CX(11)*CX(2)
         RR(J,J) = CX(12)
C
 333  GOTO 104
C
 444  RR(J,J) = S/((S*S)+16.0D0)
      GOTO 105
C
C 104  RR(J,J) = S*RR(J,J)
 104  CONTINUE
 105  CONTINUE
C
C
C
      J = JTEMP
      I = ITEMP
C
      IF (I.LT.J) THEN
         FS = RR(J,I)
      ELSEIF (I.EQ.J) THEN
         FS = RR(J,J)
      ELSE
C
      RR(1,2) = R(1,2)
      RR(2,3) = R(2,3)
      RR(3,4) = R(3,4)
      RR(1,3) = R(1,3)
      RR(2,4) = R(2,4)
      RR(1,4) = R(1,4)
      FS = RR(J,I)
C
      ENDIF
C
C     CLOSE(4,STATUS='KEEP')
      RETURN
      END
C
C
C
      COMPLEX*16 FUNCTION HYPTAN(Z)
C
      COMPLEX*16 Z
C
      DOUBLE PRECISION DN(5)
C
      DN(1) = 2*DREAL(Z)
      DN(2) = 2*DIMAG(Z)
      IF (DABS(DN(1)).GT.704) THEN
        HYPTAN = DCMPLX(1.0D0,0.0D0)
        RETURN
```

```
      ENDIF
      DN(3) = DCOSH(DN(1))+DCOS(DN(2))
      DN(4) = DSINH(DN(1))/DN(3)
      DN(5) = DSIN(DN(2))/DN(3)
      HYPTAN = DCMPLX(DN(4),DN(5))
C
      RETURN
      END
C
```

APPENDIX III

Results and Tables

To produce the complete set of results will need an additional 2,200 pages of paper. Therefore, they are provided in a set of 4 high density MSDOS formatted diskettes. The files were compressed by the PKZIP program which reduced the size of the files by about 70%. In the non-compressed form, these files requires 12MB of disk storage space.

Table 1
Samples of Difference Between AECL and Talbot Results for the source release function

RN	M	Avg Err	Max Err	Log(T_e)	f(Te)	T_1	T_2
1	73	4.000E-15	1.730E-13	4.954E+00	1.870E-11	3	6
2	73	2.401E-15	7.624E-14	3.602E+00	4.372E-10	3	8
3	73	2.731E-15	1.783E-13	3.477E+00	5.054E-10	1	6
4	73	4.374E-15	2.154E-13	4.954E+00	2.000E-11	2	8
5	73	2.376E-15	7.408E-14	3.602E+00	4.473E-10	3	7
6	73	2.731E-15	1.782E-13	3.477E+00	5.154E-10	2	5
7	73	6.519E-16	3.458E-14	3.699E+00	2.880E-09	2	1
8	73	6.519E-16	3.458E-14	3.699E+00	2.880E-09	2	1
9	73	4.817E-16	3.413E-14	3.477E+00	2.885E-09	1	2
10	73	3.805E-15	1.030E-13	5.903E+00	1.627E-11	0	5
11	73	1.204E-15	8.204E-14	4.602E+00	1.384E-10	3	7
12	73	1.045E-15	6.398E-14	4.477E+00	1.598E-10	2	10
13	73	4.476E-15	1.761E-13	5.903E+00	2.000E-11	2	5
14	73	1.174E-15	7.991E-14	4.602E+00	1.486E-10	1	7
15	73	1.041E-15	6.373E-14	4.477E+00	1.700E-10	1	6
16	73	3.337E-16	1.486E-14	4.778E+00	8.728E-10	1	3
17	73	1.000E-16	5.377E-15	4.699E+00	8.729E-10	5	2
18	73	7.744E-17	5.135E-15	4.477E+00	8.748E-10	2	2
19	73	3.236E-16	1.398E-14	5.301E+00	6.076E-11	0	7
20	73	5.610E-16	2.827E-14	5.000E+00	8.752E-11	4	6
21	73	1.345E-15	5.900E-14	4.954E+00	9.229E-11	1	3
22	73	3.031E-15	1.689E-13	6.301E+00	2.000E-11	0	8
23	73	5.460E-16	2.751E-14	5.000E+00	9.788E-11	1	3
24	73	1.335E-15	5.851E-14	4.954E+00	1.026E-10	0	2
25	73	8.949E-18	4.229E-16	5.000E+00	5.150E-10	2	6
26	73	1.984E-17	8.944E-16	5.000E+00	5.150E-10	3	0
27	73	2.148E-16	9.513E-15	5.000E+00	5.150E-10	1	1
28	73	2.923E-14	1.639E-12	2.602E+00	6.209E-09	2	10
29	73	6.348E-14	3.499E-12	3.477E+00	5.910E-10	2	8
30	73	2.272E-13	9.708E-12	1.954E+00	1.642E-08	1	3
31	73	2.933E-14	1.642E-12	2.602E+00	6.218E-09	3	10
32	73	6.276E-14	3.525E-12	3.477E+00	5.923E-10	6	9
33	73	2.272E-13	9.708E-12	1.954E+00	1.643E-08	6	5

Definitions:

RN = Test case number
M = Number of points (time) calculated
Avg Err = Average difference between the AECL and the Talbot resutls
Max Err = Maximum difference between the AECL and the Tolbot results
Log(T_e) = The time where the maximum difference occurred
f(Te) = The value of the concentration
T_1 = The number of suspect points
T_2 = The number of discrepant points

Table 2
Samples of Difference Between AECL and Talbot Results for the vault response function

RN	M	Avg Err	Max Err	Log(T_e)	f(Te)	T_1	T_2
1	60	3.898E-11	1.932E-09	0.000E+00	2.082E-03	2	1
2	54	1.094E-09	4.848E-08	0.000E+00	5.455E-02	0	1
3	43	1.059E-13	1.461E-12	-3.010E-01	1.164E-01	1	2
4	60	3.953E-11	1.959E-09	0.000E+00	2.111E-03	7	1
5	54	1.108E-09	4.912E-08	0.000E+00	5.529E-02	0	0
6	42	1.098E-13	1.479E-12	-3.010E-01	1.178E-01	1	0
7	40	5.136E-11	1.995E-09	-5.229E-01	4.288E-01	0	0
8	40	5.136E-11	1.995E-09	-5.229E-01	4.288E-01	0	0
9	44	2.704E-12	3.533E-11	-6.990E-01	1.286E-01	1	2
10	48	3.670E-12	1.497E-10	1.000E+00	5.566E-04	2	1
11	43	6.976E-13	1.411E-11	6.021E-01	1.295E-03	2	1
12	42	8.104E-15	2.813E-13	6.021E-01	4.632E-03	0	0
13	49	3.762E-12	1.567E-10	1.000E+00	5.824E-04	1	2
14	46	5.620E-13	1.243E-11	6.021E-01	1.354E-03	0	2
15	38	1.417E-14	2.936E-13	6.021E-01	4.830E-03	0	0
16	35	1.801E-12	4.347E-11	4.771E-01	2.745E-02	1	0
17	35	1.801E-12	4.347E-11	4.771E-01	2.745E-02	1	0
18	44	2.181E-13	7.831E-12	6.021E-01	1.304E-01	0	0
19	48	1.996E-09	8.713E-08	1.699E+00	5.365E-04	2	3
20	40	1.118E-13	2.738E-12	1.000E+00	3.600E-04	0	0
21	41	2.614E-15	9.599E-14	9.542E-01	5.933E-04	1	0
22	44	2.350E-09	9.404E-08	1.699E+00	5.787E-04	2	1
23	37	1.043E-13	2.337E-12	1.000E+00	3.883E-04	1	0
24	41	2.815E-15	1.033E-13	9.542E-01	6.379E-04	4	0
25	34	8.962E-12	1.717E-10	1.000E+00	1.715E-02	2	0
26	34	8.962E-12	1.717E-10	1.000E+00	1.715E-02	2	0
27	34	1.297E-13	3.965E-12	9.542E-01	2.411E-02	0	2
28	69	2.441E-08	1.099E-06	-2.000E+00	3.318E-03	4	1
29	63	8.431E-07	3.464E-05	-2.000E+00	1.047E-01	3	4
30	44	8.957E-12	2.732E-10	-2.398E+00	3.372E+01	1	1
31	70	2.407E-08	1.099E-06	-2.000E+00	3.318E-03	3	3
32	59	9.003E-07	3.465E-05	-2.000E+00	1.047E-01	1	3
33	52	7.603E-12	2.732E-10	-2.398E+00	3.372E+01	2	1
34	54	1.627E-05	5.722E-04	-2.000E+00	1.742E+00	0	4
35	54	1.627E-05	5.722E-04	-2.000E+00	1.742E+00	0	4
36	46	9.138E-12	2.921E-10	-2.398E+00	3.599E+01	0	1
37	71	3.225E-09	1.558E-07	-1.000E+00	9.596E-04	1	0
38	66	1.089E-07	4.887E-06	-1.000E+00	3.015E-02	0	5
39	51	3.056E-13	9.652E-12	-2.523E+00	9.803E-12	1	0
40	70	3.272E-09	1.558E-07	-1.000E+00	9.599E-04	5	2